DOCUMENT RESUME

ED 348 983 IR 015 663

AUTHOR Bishop-Clark, Catherine
TITLE Field Independence and Programming Achievement: A Meta-Analysis.
PUB DATE Jun 92
NOTE 23p.
PUB TYPE Reports - Research/Technical (143)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Academic Achievement; Cognitive Style; Correlation; Elementary Secondary Education; *Field Dependence Independence; Meta Analysis; Postsecondary Education; *Programing; Research Methodology

ABSTRACT

A meta-analysis is performed to determine if the cognitive style construct of field dependence/independence is related to achievement in computer programming. The studies for this meta-analysis came from three major sources: computer searches of the Educational Resources Information Center (ERIC), Psychological Abstracts, and Education Index databases; branching from bibliographies located through the computer searches; and manual searches through the last 10 years of the most relevant journals. Using the Pearson Product coefficient as the measure of effect size, 12 effect sizes from 10 studies are combined. The overall grand mean of the sample weighted effect size is .45. This suggests that there is a positive relationship between a student's degree of field independence and success in computer programming classes. Class level, programming language, and the test used to measure field independence did not significantly differentiate the correlations. Four tables and two figures display the data. The code sheet for field dependence/independence in computer programming is appended. (Contains 20 references.) (ALF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED348983

Field Independence and Programming Achievement:

A Meta-Analysis

6/16/92

Catherine Bishop-Clark
Assistant Professor
Systems Analysis Department
Miami University
Middletown, Ohio 45042
BITNET: cbclark@miavx3

Running head: FDI and Computer Programming

IR015663

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Catherine Bishop-Clark
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

Abstract

A meta analysis is performed to determine if the cognitive style construct of field dependence independence is related to achievement in computer programming. Using the Pearson Product coefficient as the measure of effect size, twelve effect sizes from ten studies are combined. The overall grand mean of the sample weighted effect size is .45. This suggests that there is a positive relationship between a student's degree of field independence and success in computer programming classes. Class level, programming language, and the test used to measure field independence measure did not significantly differentiate the correlations.

Field Independence and Programming Achievement:

A Meta-Analysis

Cognitive style is a term used to describe the systematic differences amoung individuals in perceiving and organizing their environment. Cognitive style is frequently used synonymously with learning style and is distinctly different from cognitive abilities. Style measures preferences and theoretically it is value free. In other words cognitive style A is no better than cognitive style B, it is simply different. The study of cognitive style can be traced to work as early as the 1900s. As the study of cognitive style evolved a number of instruments were designed and tested to measure different components. Examples of these different components include field dependence independence, impulsivity-reflectivity, visual-haptic, leveling-sharpening, distractibility, breadth of categorization, scanning, introversion-extroversion, analytic-holistic, and tolerance of ambiguity. Each component measures a different aspect of cognitive style and hence they are not necessarily related.

One of the most popular and best researched dimensions of cognitive style is field dependence independence (FDI). The research on this particular style originated in laboratories and dates back to the late 1940's (Witken, Moore, Goodenough and Cox, 1977). Initial research measured FDI through the Rod and Frame Test (RFT). Using this test, subjects were seated in a dark room and the image of a rod and a frame was presented. The frame was tilted and subjects were asked to adjust the rod such that it was

in a vertical position. Some subjects adjusted the rod to a vertical position regardless of the position of the frame (field independent); their perceptions were independent of the surrounding environment. Other subjects oriented the rod in relation to the tilted frame (field dependent); their perceptions were dependent on the surrounding environment (Witken et al., 1977). A second method of measuring field dependence involved a body tilting-adjusting activity. In this case, the subject was placed in a tilted room setting and then asked to adjust a movable chair such that they were sitting upright. Again, those who were able to sit in the vertical position are considered field independent, others field dependent. A third early measure of FDI was Embedded Figures Test (EFT). In this test a subject is shown a figure and then asked to locate that figure in a more complex diagram. Once again the measure is the extent which the surrounding visual framework dominates the perception of the item within it. The Group Embedded Figures Test (GEFT), was designed as a group alternative to the EFT and is the most commonly used instrument in the studies reviewed in this paper.

The study of cognitive style and its relationship to computer programming evolved from investigations attempting to predict programming aptitude. Early studies used demographic data and aptitude tests to predict programming achievement (Howell, Vincent, and Gay 1967; Mazlack 1980; Wileman, Konvalia, and Stephens 1981). When these predictors failed to explain differences in performance, researchers turned to cognitive

style. The earliest identified study which relates cognitive style to computer programming was published in 1973 and involved the construct of FDI (Testa, 1973). It seems the bulk of the studies relating cognitive style to computer programming was published in the mid 1980's. The studies performed in the early and mid 80s may have been a response to Ben Schneiderman's (1980) call to begin investigating cognitive style as one of the factors that may explain the large variation in programming ability.

There are two kinds of studies which relate FDI to computer programming. First, there are several studies which investigate if the activity of computer programming will change cognitive style (Cathcart, 1990; Vaidya, 1985). The question asked in these studies is if experience in computer programming will cause a person's cognitive style to become more field independent. The problem with these studies is that there appears to be an assumption that certain cognitive styles (in this case field independence) are better than others. The treatment in these experiments is computer programming instruction and the post-test is a 2nd measure of cognitive style. It appears the goal of the treatment is to adjust or modify cognitive style, a notion foreign to the theory. The second kind of study is one which correlates the FDI construct to success in computer programming. The question asked in these studies is if a tendency toward field independence is related to achievement or success in computer programming. The goal of this paper is to summarize studies relating to the second question. That is, does the cognitive

style construct of FDI relate to achievement in computer programming.

Although there have been previous reviews on the construct of field dependence independence in general (Witken et al., 1977), this is the first review of the research relating the construct specifically to computer programming. FDI theory suggests that those who are able to impose structure on seemingly unstructured situations (field independent) will have greater success in the activity of computer programming.

Method

Details of the Literature Search

The studies considered for use in this meta-analysis came from three major sources. The first group of studies came from computer searches of Education Resources Information Center (ERIC) (1966-1979, 1980-1991), Psychological Abstracts (1983-1991), and the Educational Index (1983-1991). The keywords used in the search were Field Dependence Independence, Computer Science Education, Information Sciences Education, and Programming. The second set of studies were retrieved by branching from the bibliographies located through the computer searches. A third set of studies came from manual searches through the last ten years (1981-1991) of the most relevant journals. Journals in the manual search included the Journal of Research on Computers in Education, Journal of Educational Computing Research, Communications of the ACM, and SICGSE Bulliton.

Criteria for Relevance

Several criteria were established for inclusion of studies into the present meta-analysis. First, studies had to relate the construct of FDI to achievement in computer programming. Second, the studies had to provide quantitative data (typically in the form of correlations) and third, the studies had to be retrievable in a eight week period. Studies which related FDI to other aspects of computers such as computer literacy or success in using computer based tools (word processors, spreadsheets, etc.) were not included in the review. Similarly, studies which measured change in FDI as a result of programming experience were not included. A total of 10 studies met all of the criteria for relevance.

Prototypical Methodologies

The majority of the studies included in this review followed a similar format. Typically, students were given a test to measure FDI at the beginning of class (whether the class be a university course or a elementary school workshop), then participated in a computer programming class, and last completed a post test designed to measure programming achievement. The students FDI score was then correlated with achievement in computer programming activities and was reported most commonly as a Pearson Product Coefficient.

Coding Categories

The coding sheet is included in Appendix A and is divided into seven categories. The first category includes the publication features of the paper. The second and third categories includes information regarding the setting of the research and the characteristics of the treatment. Section four identifies which test was used to identify FDI and section five outlines the dependent measure. The last section includes information regarding the actual statistical outcome of the test.

Operational Definitions

FDI was measured most often by the Group Embedded Figures Test and the Embedded Figures Test. In two cases the Hidden Patterns test was used to assess FDI and in one case Barkin's Instrument was used. The dependent variable was achievement and was measured in several ways. In some cases the dependent measure was reported as either a final class grade or a final exam grade. In other cases, components of the class grade were given. Examples of such components include debugging, syntax, program comprehension, and program modification scores. When several measures of achievement were given, the measures were combined and treated as one dependent measure. The rational for the combination is that final exam grades or final class grades typically consist of the very components that were measured independently in several studies.

Unit of Analysis

The shifting unit of analysis (r-index) for each sample was identified. In studies where there was one correlation reported on one sample size, the correlation for the study was the effect size. In studies where several correlations were reported (debugging score, syntax score, logic score), the correlations were averaged and the average correlation was used as the effect size. In studies that included more than one sample group (for instance correlations were reported for Sophomores and Seniors), each sample contributed one effect size. In short, each sample contributed to the general finding. This strategy allows studies to retain their information value while minimizing the violations to the assumption of independence of hypothesis tests (Cooper, 1989).

Reporting Effect Size / Statistical Test Used

Although some authors report that the actual technique used to average correlations makes no practical difference (Glass, 1977), the author used Fisher's z-transformation as suggested by Cooper (1989) to combine effect sizes. Using this technique correlations are converted to z scores, z scores are combined using a weighted average, and the computed average is then transformed back to a correlation. Hedges and Olkin (1985) provide a method for adjusting r-indexes for bias but they explain that the biases of correlations is seldom of practical concern when the sample sizes are greater than 15. In this particular review, all samples are larger than 15; therefore,

there was no adjustment for bias. Both confidence intervals and homogeneity analysis was performed using the z transformation technique.

Validity Although publication bias can be one of the primary threats to the validity of meta-analysis, the author does not perceive this to be a significant problem for this review. Many of the studies correlated a number of constructs (demographic, programmer aptitude tests, and cognitive styles) to success in computer programming with the cognitive style dimension of FDI being only one of many. Hence, in many cases the majority of the factors were non-significant. Several other articles actually tested a hypothesis that was unrelated to the cognitive style. The FDI correlation was an incidental reporting that we were able to incorporate in this review. A second threat to validity is that all reports were coded by one individual, hence there is a the potential for error due to the lack of inter-rater reliability scores.

Results

A total of 10 studies and 12 effect sizes were analyzed. Table 1 summarizes the study, effect sizes, and several of the more relevant study features. Figure 1 gives the reader a sense of the distribution of subjects, the computer language, and the instruments used to measure FDI. As weighted effect sizes were used to compute the overall effect sizes, figure 1 shows the percentage of students falling in each category. Figure 1 reveals that 64.7% of the students in the review were in

university programming classes and 35% in K-8. With regard to the computer language, 36.3% of the students used LOGO, 11.6% used Pascal, 30.7% used BASIC, 12.8% COBOL, and 9.6% Fortran. All of the elementary and junior high schools used LOGO whereas all University courses used the more business oriented languages such as COBOL and BASIC. The majority of students (65.5%) used the GEFT to measure FDI: 12.2% used the Hidden Patterns test, 16,4% used the EFT and 7.0% used Barkin's Instrument.

Table 1: Study Features

Study Characteristics

Study	N	DM	r-index	Grade	Language	Instrument
Bradley	26	1	0.63	elem	LOGO	Hidden Patterns Te
Cavaini	30	5	0.29	univ	COBOL	GEFT
Cheney	35	1	0.82	univ	BASIC	Barkin Instrument
Foreman	46	4	0.39	univ	BASIC	GEFT
Hassell-a	29	2	0.07	univ	Fortran	EFT
Hassell-b	19	2	0.49	univ	Fortran	EFT
Stevens	73	1	0.35	univ	BASIC	GEFT
Testa	34	1	0.80	univ	COBOL	EFT
Webb	35	5	0.45	jrhi	LOGO	Hidden Patterns Te
Werth	58	1	0.32	univ	Pascal	GEFT
Wilson-a	58	1	0.47	elem	LOGO	GEFT
Wilson-b	58	1	0.33	elem	LOGO	GEFT

N: Number of subjects
DM: Number of Dependent Measures

Figure 1: Distributions of Subjects

Class Level Programming Language Testing Instrument

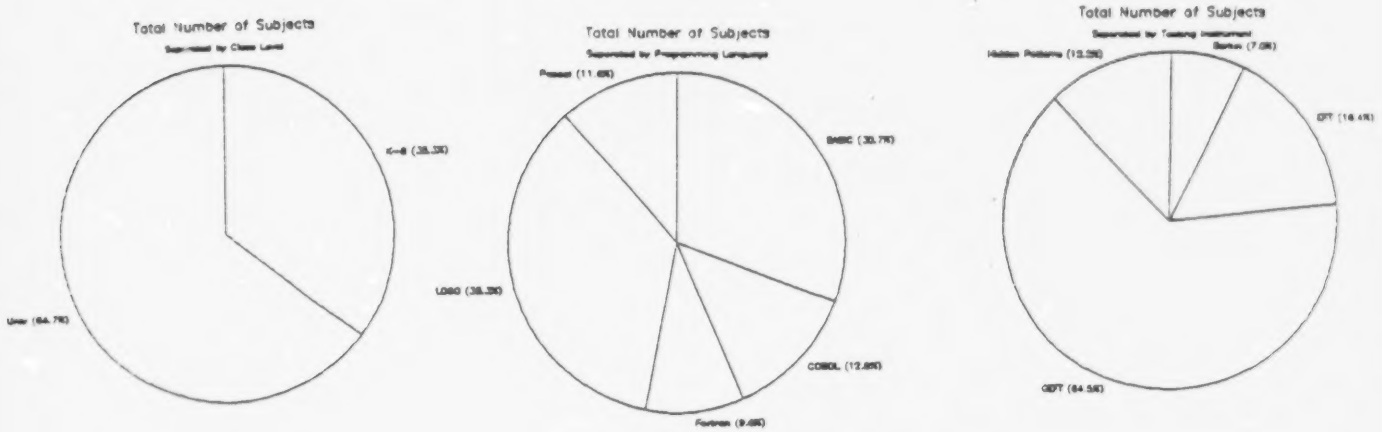
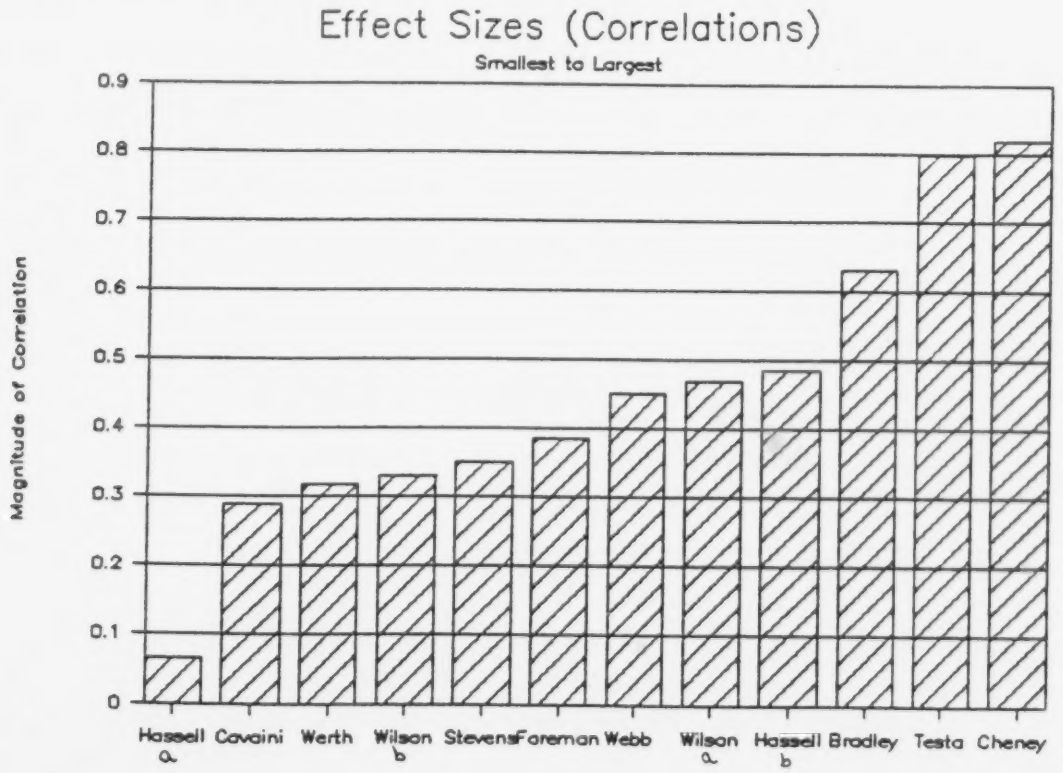


Figure 2 graphically illustrates the magnitude of the correlations collected for each sample. Although the correlations vary considerably (.08 to .82), all of the correlations are positive. That is in all studies the more field independent a student tested, the higher their programming achievement score. Cohen labels an effect size as small if the correlation = .1, medium if r = .3 and large if r = .5 (Light, Singer & Willett, 1990). Using these definitions, the first study in table 1 yields a small effect size and the last three yield large effects. The remaining 8 studies report medium effect sizes. The overall weighted correlation used Cooper's (1989) r to z transformation method is .45. The confidence interval using this method is (.395, .532). As zero is not in the confidence interval we can confidently reject the null hypothesis that there is no relation between FDI and programming achievement. The homogeneity of variance test for the 12 studies yielded a test statistic of 27.12 and a critical value at the .05

level of 18.3. From these values, we conclude that sample error alone does not explain the differences in correlations.

Figure 2: Correlations for Each Study



Tables 2, 3 and 4 show the weighted average effect sizes when the effects are separated by programming language, testing instrument, and grade level. Note that in all of the tables only one confidence interval crosses zero (Fortran). In all other cases the confidence interval does not cross zero. Homogeneity of variance was found when Fortran was the programming language, when LOGO was the programming language, when GEFT was the test instrument, when the hidden patterns test was the test instrument

and when K-8 was the grade level. One-way Anovas were run on SPSSX using effect size as the dependent variable to see if any significant differences existed between programming languages. No significant differences were found. Similarly, no significant differences were found amoung testing instruments or class levels. Note that with only 12 effect sizes, the power of the Anova is extremely poor and even if a difference exists, the likelihood of picking up a significant difference is extremely small.

Table 2:

Weighted Average Effect Size
and Homogeniety of Variance Results
Separated by Programming Language

Program Language	d-level	95% Conf Interval	No of d-levels	No of Subjects	Homogeniety	Chi Sq .05	
BASIC	0.46	(0.33 0.57)	3	154	6.52	5.99	
COBOL	0.62	(0.45 0.75)	2	64	9.24	3.84	
Fortran	0.24	(-0.01 0.48)	2	48	2.14	3.84	*
LOGO	0.51	(0.40 0.61)	4	177	1.09	9.35	*
Pascal	0.32	(----NA------)	1	58	NA	NA	

* Homogeniety of Variance test is accepted

Table 3:

Weighted Average Effect Size
and Homogeniety of Variance Results
Separated Testing Instrument

FDFI Test	d-level	95% Conf Interval	No of d-levels	No of Subjects	Homogeniety	Chi Sq .05	
Barkin	0.68	(----NA------)	1	35	NA	NA	
EFT	0.44	(0.37 0.68)	3	82	15.23	5.99	
GEFT	0.39	(0.30 0.33)	6	323	2.06	11.1	*
Hidden Pa	0.59	(0.40 0.73)	2	61	0.20	3.84	*

* Homogeniety of Variance test is accepted

Table 4:

Weighted Average Effect Size
and Homogeniety of Variance Results
Separated by Class Level

FDFI Test	d-level	95% Conf Interval	No of d-levels	No of Subjects	Homogeniety	Chi Sq .05	
K-8	0.51	(0.40 0.57)	4	177	1.08	7.81	*
Univ	0.47	(0.35 0.40)	8	324	25.09	14.10	

* Homogeniety of Variance test is accepted

Discussion

This review reveals that students who test as field independent consistently earn higher grades in educational computer programming classes than their field dependent counterparts. All 12 effect sizes reported in this study were positive and the overall weighted effect size was .45. Taken individually, the studies in the report generally have a low sample size (average n = 42) and are difficult to generalize. However, the consistency of a positive relationship amoung different languages, different age groups, and different measures of FDI makes it difficult to assert that there is no relation between the cognitive style of FDI and computer programming. Although we cannot conclude that field independence causes higher

grades; the studies reviewed clearly reveal that field independence is positively related to achievement in a computer programming classes.

As this is the first review of the construct of FDI and its relationship to computer programming, it is not possible to compare these results with other reviews. It is however possible to relate these findings other logical reasoning activities. The FDI construct has been associated with logical reasoning ability (Witkens et al., 1977) and the theory predicts that field independent students will perform better in analytic fields than field dependent students. This review further confirms this prediction.

One of the limiting conditions of the review is the small number of effect sizes in the analysis. Only 12 effect sizes are used and when subdivided by programming language and testing instrument, the number of effects per cell is as small as 1. The very small number of effect sizes per cell makes both the homogeneity of variance and the results of the ANOVAs suspect. Additionally, all of the studies in this review that were in the K-8 grade level used LOGO as the programming Language. Therefore, it is not possible to investigate if differences are due to programming language or if they are due to grade level.

One of the implications that can be drawn from this review is that field independent students may need some additional help to succeed in computer programming courses. Tools such as structure charts, pseudocode, and flowcharts may be especially

useful for the field dependent students as the purpose of the tool is to impose an initial structure on an unstructured situation. Testa (1973) suggests that the FDI score should be used as a factor in assessing programming potential. In other words, he suggests using the FDI score as a factor to advise students whether or not to enter the programming field. The author is very skeptical about such a strategy. As mentioned above although a relationship exists, it is not necessarily casual! Perhaps the most important implication of the research is that these studies give researchers some additional insight to the enormous variability that exists in the achievement of the student programmer.

References

Bradley, C. (1985). The Relationship Between Students' Information Processing Styles and LOGO Programming. Journal of Educational Computing Research, 1(4), 427-433.

Cathcart, W. (1990). Effects of LOGO Instruction on Cognitive Style. Journal of Educational Computing Research, 6(2), 231-242.

Cavaini, T.P. (1989). Cognitive Styles and Diagnostic Skills of Student Programmers. Journal of Research in Computing in Education, 21, 411-420.

Cheney, P. (1980). Cognitive Style and Student Programming Ability: An Investigation. AEDS Journal, 13, 285-291.

Cooper, H., Integrating Research: A Guide for Literature Reviews, Sage Publications, Inc., 1989.

Foreman, K. (1988). Cognitive Style, Cognitive Ability and Acquisition of Initial Programming Competence. Paper presented at the Annual Meeting of the Association for Educational Communications and Technology, New Orleans, LA, Jan 14-19, 1988. (ERIC Document Reproduction Service No. ED 295 638).

Glass, G. (1977). Integrading Findings: The Meta-Analysis of Research. In L. Shulman (Ed.) Review of Research in Education, 5, 351-379.

Hassell, J. (1982). Cognitive Style and a First Course in Computer Science: A Success Story. AEDS Monitor, 21, 33-35.

Hedges, L. & I. Olkin (1985). Statistical Methods for Meta-Analysis. Orlando, Florida: Academic Press, Inc.

Howell, M., Vincent, J., and R. Gay. (1967). Testing Aptitude for Computer Programming. Psychological Reports, 20, 1251-1256.

Light, R., Singer, J., and J. Willett (1990). By Design Cambridge, Massachusetts: Havard University Press.

Mazlack, L. (1980). Identifying Potential to Acquire Programming Skill. Communications of the ACM, 23, 14-17.

Stevens, D. (1983). Cognitive Processes and Success of Students in Instructional Computer Courses. AEDS Journal, XX, 228-233.

Testa, C. (1973). A New Approach for Programming Aptitude Testing. Proceedings of the 11th Annual Conference of the Special Interest Group for Computer Personel Research, 11, 49-61.

Vaidya, Shelia Rao. Individual Differences Amoung Young Children in LOGO Environments. Computers and Education, 9(4), 221-226.

Webb, N. (1984). Microcomputer Learning in Small Groups: Cognitive Requirements and Group Processes. Journal of Educational Psychology, 76(6), 1076-1088.

Werth, L. (1986). Predicting Student Performance in a Beginning Computer Science Class. ACM-SIGSCE Bulletin, 18(1), 138-142.

Wileman, S., Konvalina, J., and L. Stephens (1974). Factors Influencing Success in Beginning Computer Science Courses. Journal of Educational Research, 74(4), 223-226.

Wilson, D., Mundy-Castle, A. and P. Sibanda (1990). Field Differentiation and LOGO performance amoung Zimbabwean Schoolgirls. Journal of Social Psychology, 130(2), 277-279.

Witken, H, Moore, C., Goodenough, D., and P. Cox (1977). Field-Dependent and Field-Independent Cognitive Styles and Their Educational Implications. Review of Educational Research, 47(10), 1-64.

Appendix A

Code Sheet for Field Dependence / Independence in Computer Programming

I. SOURCE OF REFERENCE / PUBLICATION FEATURES

Authors:

Title:

Journal:

Year: Volume: Pages:

Source of Reference:

(ERIC, Psych Abstract, Education Index, Grandfathering, ABI)
(Dissertation Abstracts, Computer and Information Systems Index)
(Computer Literature Index)

II. SETTING OF THE RESEARCH

Sample Size:

___ # of schools ___ # of classrooms
___ # of students
___ # of FI students ___ # of FD students

Student Variables:

_____ Grade Levels (pre-elem, elem, JH, HS)
University (Fr, So, Jr, Sr, Grad)
_____ SES: (low, middle, upper, NG)
_____ Ethnicity: (black, white, Hispanic, composit, NG)
_____ Ability Level (low, average, high, composit, NG)

School Variables:

_____ Location (use state abbreviations)
_____ Type (urban, suburban)

III. Characteristics of the Treatment

Computer Language: Length of Programming Class

____ LOGO ___ Number of weeks
____ Pascal ___ Hours per week
____ BASIC
____ COBOL
____ Fortran

Computers per Students:
___ 1:1 ___ 1:3
___ 1:2 ___ 1:4 and up

IV. Independent Variable (Measuring Instrument):

___ Group Embedded Figures Test
___ Embedded Figures Test
___ Hidden Patterns Test
___ Barkin's Instrument

V. Dependent Variable (Achievement):

___ Teacher Ratings
___ Final Exam Grade
___ Class Grade
___ Program Comprehension
___ Program Composition
___ Program Modification
___ Program Debugging
___ Syntax
___ Other Explain_____

VI. Statistical Outcome:

Pearson Product Correlation

Teacher Ratings_____
Final Exam Grade _____
Class Grade_____
Program Comprehension _____
Program Composition _____
Program Modification _____
Program Debugging_____
Syntax _____
Other_____

Other Statistical Measures:

_____
_____
_____
_____

Overall Effect Size:

VII. Other Notes and Comments:

_____
_____
_____
_____